Syllabus CS 350 – Fall 2020

Introduction to Software Engineering

Instructor: Jay Morris

Time: Online meetings to be arranged, recitations as scheduled.

The course will introduce students to a wide variety of software packages. All of these are open-source, free software, but students will need to install some of these on their chosen development machine (whether their own PC or in their account on the CS network).

Students will need frequent access to a PC capable of hosting software development activities or of connecting remotely to CS Dept servers where such activities can be performed.

Students will be attending network conferences requiring the use of a microphone. Webcams are optional.

For both remote access to CS Dept servers and for network conferencing, a good-quality internet connection is important.

Project review sessions will be scheduled for selected weeks during the recitation periods. Attendance at these is mandatory. Failure to attend will result in substantial grade penalties for that portion of the project.

Attendance at other scheduled recitation sessions, as announced in the course outline, is also mandatory. Failure to attend may result in substantial grade penalties.

Topics & Learning Objectives

Students completing this course should be able to:

- Demonstrate an understanding of the overall strategy of software development:
  - Discuss the phases and component activities of software development
  - Assess the likely impact of popular software process development models on a project.
  - Discuss common team organizations and roles in software development.
- Work with software requirements documents
  - Read common forms of requirements documents
  - Write at least one standard form of requirements document
  - Apply requirements to guide the subsequent construction of software
Apply best practices in collaborative software construction
  o Discuss the issues and problems involved in collaborative development of software.
  o Evaluate the suitability of alternative best practices for a software construction project.
  o Support common best practices of via a modern IDE and associated tools
  o Apply a variety of software measurement and estimation techniques.

Due Dates and Late Submissions

Late assignments and make-up exams will not normally be permitted.

Exceptions to this and other grading policies will be made only in situations of unusual and unforeseeable circumstances beyond the student’s control. Arrangements must be made prior to the due date in any situations where the conflict is foreseeable.

“I’ve fallen behind and can’t catch up”, “I’m having a busier semester than I expected”, or “I registered for too many classes this semester” are not grounds for an extension.

Academic Honesty

Everything turned in for grading in this course must be your own work, or, for team projects, the work of your own team. Opportunities for teamwork will be clearly identified as such.

Students are expected to conform to academic standards in avoiding plagiarism.

  • Among other things, this means that if you use ideas found on the internet (outside of the course website) in your answers to an assignment or exam question (including when coding!), you must cite your sources appropriately.

    If you use text directly taken from such sources you must appropriately designate the quoted material as such.

The instructor reserves the right to question a student orally or in writing and to use his evaluation of the student’s understanding of the assignment and of the submitted solution as evidence of cheating.

Students who contribute to violations by sharing their code/designs with others may be subject to the same penalties. Students are expected to use standard Unix protection mechanisms
(chmod) to keep their assignments from being read by their classmates. Failure to do so will result in grade penalties, at the very least.

This policy is *not* intended to prevent students from providing legitimate assistance to one another. Students are encouraged to seek/provide one another aid in learning to use the operating system, in issues pertaining to the programming language, or to general issues relating to the course subject matter.

Student discussions should avoid, however, explicit discussion of approaches to solving a particular programming assignment, and under no circumstances should students show one another their code for an ongoing assignment, nor discuss such code in detail.

Violations of this policy will be reported to the Office of Student Conduct and Academic Integrity for consideration for punitive action.

**Grading**

<table>
<thead>
<tr>
<th>Assignment</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual Assignments</td>
<td>15%</td>
</tr>
<tr>
<td>Semester Project</td>
<td>45%</td>
</tr>
<tr>
<td>Midterm Exam</td>
<td>15%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>25%</td>
</tr>
</tbody>
</table>

Individual Assignments (15% in total) and individual Semester Project Phases (50% in total) will not necessarily be equally weighted.

Grading is [normalized](#).

**4 ASSIGNMENT GRADING**

Individual assignments will be turned in through the CS submission system, rather than through Blackboard–more information is available [here](#). Most of the assignments will be graded by an automatic grader. The results will be sent to your email account. Unless the assignment explicitly states otherwise, you may submit a total of three times; the instructor will take the last of the marks, although you may request that your score be “rolled back” to an earlier one. You may NOT submit after viewing the sample solution. Students will engage in team projects in this course. Students are expected to actively participate in and contribute to their teams, and this engagement with the team will be part of the grade.